Gluon Smart Contract Review

The review focused on the following contract:GluonWBoxGuardScript. Below is a summary of findings, categorized into components, along with additional comments.

Component	Agree	Disagree
Contracts seem to match intended functionality	Mostly Agree*	
Contracts seem to have no potential compilation errors	Agree	
All expected outputs seem correctly formed	Agree	
All evaluations seem to be correct	Agree	
There does not seem to be any vulnerabilities	Agree	
There does not seem to be any potential overflow errors	Agree	
It does not seem possible to create fraudulent oracle data	Agree	

General Comments:

The contract is well-structured and styled. It defines clear spending paths and demonstrates careful handling of potential overflow errors. The contract seems to match the requirements of the project. All variable definitions are logical and purposeful. No potential draining exploits were found.

Specific Comments:

Assumption: If the purpose of *isUpdateTreasury* is to update the multisg address and recreate the UTXO as is, one additional validation is missing:

SELF.propositionBytes == OUTPUTS(0).propositionBytes

Impact: This does not create a public exploit, however the omission of it does allow the multisig to drain the contract entirely. This may be desired in beta stage, but to become trustless it should be added.

Additional Notes (No Actions Needed):

Enforcement of nDays <= Buckets:

The cap of *nDays* = *BUCKETS* is enforced only in the *betaDecayMinus* transaction and not in *betaDecayPlus*. Despite this, there does not appear to be a way to exploit this behavior. The contract will execute correctly even when *n* > *BUCKETS*, as it ensures that the volumes array is set to all zeros for indices greater than 0 and array size length is enforced.

Handling of negative VarPhiBeta:

Note that if the volume calculated is twice the value of R that VarPhiBeta becomes negative, however as discussed on Discord it does not seem possible to generate such a volume, also even if it were possible this can only become exploitable if the fusionRatio is also below 0, which is not possible.

Use of Input States for R, Sn, Sp:

In discussions with Bruno, the use of input states for R, Sn, and Sp was confirmed to be an intentional design choice. While such an approach can sometimes introduce risks like wholesale attacks, discounts, or manipulation in other contexts, this contract's specific implementation appears secure. After simulating various scenarios, no profitable exploitation paths could be identified. Therefore, this design decision is deemed safe and appropriate for the contract.

Checks Regarding Protocol Formulas:

Do also note that in this review I did not look to question the validity of the formulas provided in the Gluon W paper, instead I simply checked that the contract was a valid implementation of the formulas provided. As I concluded in my review, I believe the implementation to be valid.